

WORDCRAFT – ALUSTARIIPPUMATON SOSIAALINEN MONINPELI

Sami Kostilainen

Opinnäytetyö
Helmikuu 2013

Ohjelmistotekniikka
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) KOSTILAINEN, Sami	Julkaisun laji Opinnäytetyö	Päivämäärä 15.02.2013
	Sivumäärä 22	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkkojulkaisulupa myönnetty (x)
Työn nimi WORDCRAFT – ALUSTARIIPPUMATON SOSIAALINEN MONINPELI		
Koulutusohjelma Ohjelmistotekniikan koulutusohjelma		
Työn ohjaaja(t) LAPPALAINEN-KAJAN, Tarja		
Toimeksiantaja(t) Star Arcade Oy		
<p>Tiivistelmä</p> <p>Opinnäytetyönä tehtiin WordCraft-niminen sosiaalinen, alustariippumaton moninpeli toimeksiantona jyväskyläiselle peliohjelmointiyritykselle Star Arcadelle.</p> <p>Työ toteutettiin C++ ohjelmointikielellä ja pelin toteutuksessa käytettiin Star Arcaden DieselEngine-pelimoottoria, sekä sosiaalista StarArcadeClient-moninpelikehystä.</p> <p>Pelin ideana on muodostaa annetusta kirjainkasasta monimutkaisia ja harvinaisia sanoja mahdollisimman nopeasti. Peliä pelataan verkkopelinä toista pelaajaa sekä aikaa vastaan identtisillä kirjainkasoilla. Pelaamalla hyvin pelaaja voi vaikuttaa vastustajan pelin kulkuun jäädyttämällä vastapelaajan kirjainkasaa. Pelissä käytetään englanninkielistä sanakirjaa löydettyjen sanojen tunnistamiseen.</p> <p>Pelin dokumentointi muodostui alustariippumattoman kehityksen eri osa-alueista, katselmoinnista pelin kehityksessä käytettyihin menetelmiin ja työkaluihin, toteutuksen ja työprosessin eri vaiheista, sekä pohdinnasta, kuinka pelin kehitys onnistui kokonaisuudessaan.</p> <p>Vaikka pelin kehityksessä ei ilmennyt suuria ongelmia, ei peliä ole opinnäytetyön tekohetkellä julkaistu, vaan se julkaistaan vuoden 2013 ensimmäisellä neljänneksellä. Ennen julkaisua pelistä muutetaan käyttöliittymä ja nimi, ja se toteutetaan uudempaan sosiaaliseen moninpelikehykseen.</p>		
Avainsanat (asiasanat) Peliohjelmointi, C++		
Muut tiedot		



Author(s) KOSTILAINEN, Sami	Type of publication Bachelor's Thesis	Date 15.02.2013
	Pages 22	Language Finnish
	Confidential () Until	Permission for web publication (x)
Title WORDCRAFT – SOCIAL CROSS-PLATFORM MULTIPLAYER GAME		
Degree Programme Software Engineering		
Tutor(s) LAPPALAINEN-KAJAN, Tarja		
Assigned by Star Arcade Oy		
<p>Abstract</p> <p>This thesis discusses social, cross-platform multiplayer game called WordCraft and it was assigned by a game development company, Star Arcade in Jyväskylä.</p> <p>The study was implemented in C++ programming language, and was carried out with DieselEngine – the game engine used in Star Arcade, and StarArcadeClient – social multiplayer layer.</p> <p>The idea of the game is to form complex and rare words from the given letter stack as quickly as possible. The game is played online against another player, as well as against the clock with identical letter stacks. By playing well the player can affect the game play by freezing the other player's letter stack. The game uses an English dictionary for recognizing the words.</p> <p>The documentation of the game consists of different areas of cross-platform development, review of the methods and tools used in development and the various stages of the working process, as well as speculation on how successful the game development was as a whole.</p> <p>Although the development of the game was a success, the game has not been released at the time of writing the thesis, however, it will be released in the first quarter of 2013. The name and the user interface of the game will be changed before the initial release, and it will be implemented to a newer social multiplayer layer.</p>		
Keywords Game programming, C++		
Miscellaneous		

SISÄLTÖ

1	TYÖN LÄHTÖKOHDAT	3
1.1	Tehtävät ja tavoitteet	3
1.2	Star Arcade Oy	3
1.3	WordCraft pelinä	4
2	ALUSTARIIPPUMATON KEHITYS	8
2.1	Määritelmä	8
2.2	Hyödyt ja haasteet	9
2.3	Käytetyt alustat ja niiden eroavaisuudet	10
3	MENETELMÄT JA TYÖKALUT	11
3.1	Ohjelmointikieli	11
3.2	Työkalut	11
3.3	OpenGL ja OpenGL ES	12
4	TOTEUTUS JA TYÖPROSESSI	13
4.1	Aikataulu	13
4.2	StarArcadeClient	14
4.3	DieselEngine	15
4.4	WordCraft	15
4.5	Tekoäly	19
5	POHDINTA	20

5.1	Johtopäätökset	20
5.2	Tulevaisuus	20
LÄHTEET		22

KUVIOT

KUVIO 1.	WordCraftin pelitila	7
KUVIO 2.	Pseudokoodiesimerkki alustariippuvaisesta ohjelmakoodista	8
KUVIO 3.	Pseudokoodiesimerkki alustariippumattomasta ohjelmakoodista	9
KUVIO 4.	Ikkunan alustaminen GLUT rajapintaa käyttäen	13
KUVIO 5.	Pelitilan rakenteen koodiesimerkki.....	16
KUVIO 6.	WordCraftin luokkahierarkia	18

TAULUKOT

TAULUKKO 1.	Kirjainten pisteiden ja jakojen suuruudet	6
-------------	---	---

1 TYÖN LÄHTÖKOHDAT

1.1 Tehtävät ja tavoitteet

Opinnäytetyönä tehtiin alustariippumaton sosiaalinen moninpeli WordCraft toimeksiantona jyväskyläiselle peliohjelmointiyritykselle Star Arcadelle. Alustariippumattomuus kattaa Star Arcaden tukemat alustat, joihin kuuluvat mm. iOS, Android ja Windows Phone 7. Sosiaalisena puolena pelistä löytyvät mm. kaverilistat ja pikaviestien lähettäminen. Peliä pelataan verkossa toista pelaaja vastaan.

Työ toteutettiin kolmen hengen projektina johon kuului ohjelmoija, graafikko ja projektipäällikkö. Toteutuksen lisäksi kehityksessä mukana olivat Star Arcaden testausryhmä sekä johtoryhmä.

Pelin valmistumiselle ei varsinaista aikarajaa ole asetettu ja peliä ei ole vielä julkaistu dokumentoinnin kirjoitushetkellä, sillä pelin tekoon ja julkaisuun liittyy paljon muutakin kuin ohjelmakoodin kirjoittaminen.

1.2 Star Arcade Oy

Star Arcade Oy on vuonna 2010 perustettu pelejä valmistava yritys. Se keskittyy mobiilipelien kehittyville markkinoille ja työllistää noin 30 ihmistä. Päätoimisto sijaitsee Jyväskylässä. Yrityksen tuotteisiin kuuluvat laitteistoriippumattomat, sosiaaliset moninpelit. (Company Presentation 2012.)

Yrityksen markkinointistrategia toimii freemium-mallilla, joka tarkoittaa ilmaisia pelejä, joissa on mahdollista ostaa virtuaalisia esineitä oikealla rahalla. Esineitä voi myös lahjoittaa kavereille riippumatta siitä, millä alustalla he pelaavat. (Mt.)

Star Arcade Oy:n pelit ovat reaaliaikaisia, ja niitä on saatavilla yli 180 maassa mm. App Storesta, Google Playsta, Facebookista ja monesta muusta sisältöpalvelusta. Jokaisessa pelissä on mukana sosiaalinen kerros, joka mahdollistaa mm. pelaajien pisteiden seurannan, keskustelun ja kaverilistat. On myös mahdollista järjestää operaattorikohtaisia turnauksia peleihin. (Mt.)

1.3 WordCraft pelinä

WordCraft on Scrabble-henkeen perustuva kaksin pelattava sanapeli, jossa pelataan toista pelaajaa, aikaa ja omia ennätyksiään vastaan (ks. kuvio 1). Pelin pääsuunnittelijana ja graafikkona toimi Jukka Korhonen ja kehittäjänä Sami Kostilainen. Pelitestauksen ja pelin tasapainottelun hoiti kehittäjän ja suunnittelijan lisäksi Star Arcaden testausosasto.

Pelin ideana on muodostaa mahdollisimman pitkiä ja harvinaisia englanninkielisiä sanoja annetusta kirjainkasasta niin pian kuin mahdollista. Mikäli muodostettu sana on tarpeeksi arvokas, saa sanasta palkinnoksi kertoimen, jonka voi käyttää myöhemmissä sanoissa nostaakseen saatua pistemäärää. Pelin tasapainon vuoksi sanoista, jossa on käytetty kerroin, ei voi saada uutta kerrointa.

Pelissä on myös mahdollista vaikuttaa vastassa olevan pelaajan peliin. Muodostamalla pitkän sanan pelaaja lähettää vastustajalleen lumihiutaleita, jotka jäädyttävät vastustajan peliruutua. Vastustajan tulee sulattaa jäät "näpsyttämällä" ruutua jäiden kohdalta, ennen kuin voi valita kirjaimia jään alta.

Pelissä käytetään englanninkielistä sanakirjaa. Sanakirjan sisältö perustuu lukuisten eri englanninkielisten sanakirjojen yhdistämiseen. Sanakirjaan hyväksytään vain nimissään kolmen ja maksimissaan 15 kirjaimen sanat.

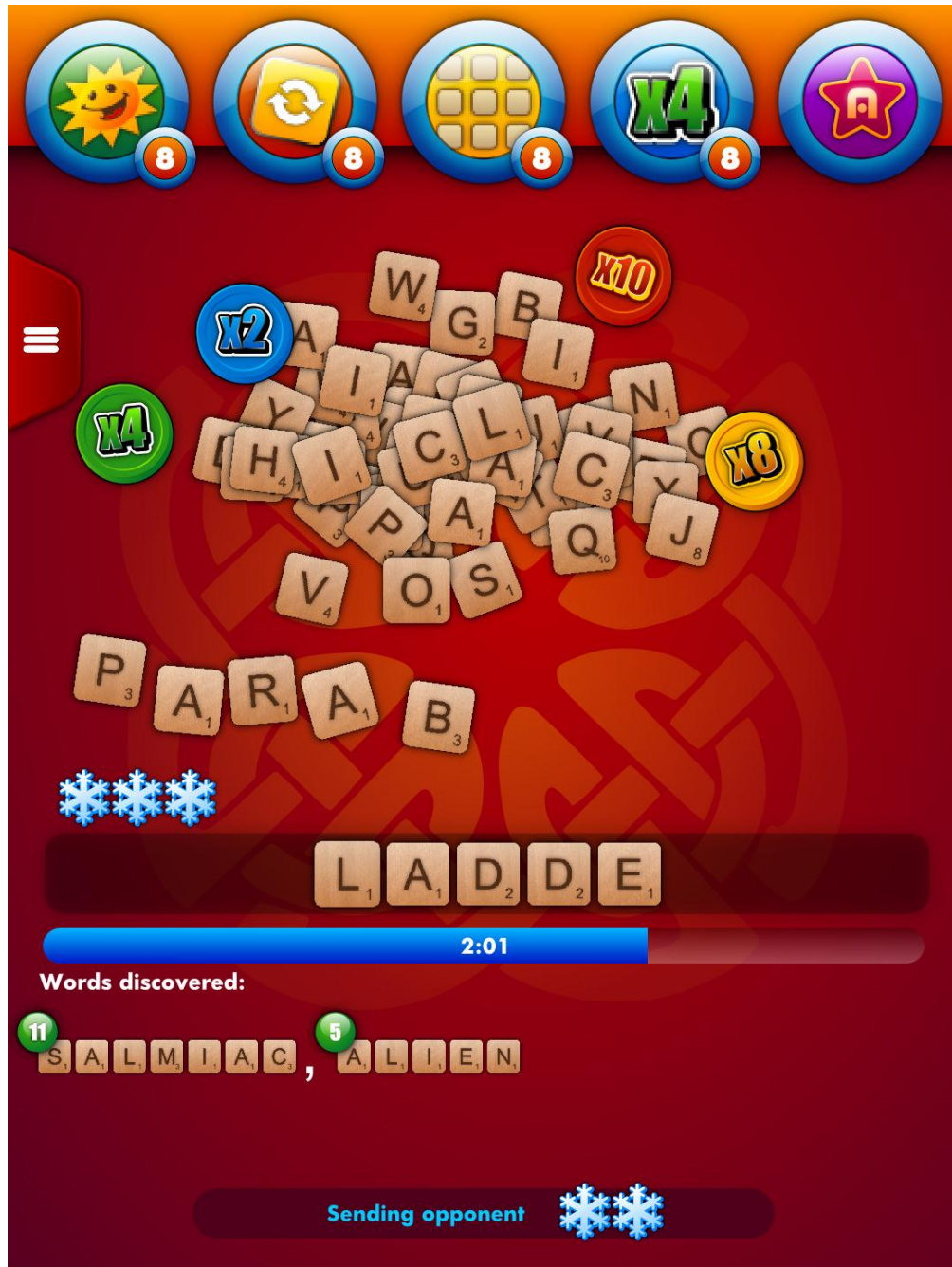
Kirjainten harvinaisuus perustuu kirjainten arvokkuuteen, eli jokainen kirjain on ennalta määrätyn pistemäärän arvoinen. Muodostuneen sanan arvo on kirjainten arvojen yhteissumma kerrottuna mahdollisesti käytetyllä kertoimella.

Kirjainkasan kirjainten jako on toteutettu käyttämällä jakoperiaatetta kirjainten sanassa esiintymisen harvinaisuuden mukaan. Peli muodostaa edellä mainittua periaatetta käyttäen 200 kirjainta, joista se ottaa satunnaisesti 100 pelissä käytettävään kirjainkasaan (ks. taulukko 1). Peli käyttää identtistä kirjainkasa molemmille pelaajille.

TAULUKKO 1. Kirjainten pisteiden ja jakojen suuruudet

Kirjain	Pistemäärä	Jako %
A	2	9
B	6	2
C	6	2
D	4	4
E	2	12
F	8	2
G	4	3
H	8	2
I	2	9
J	16	1
K	10	1.5
L	2	4
M	6	2
N	2	6
O	2	8
P	6	2
Q	20	1.5
R	2	6
S	2	4
T	2	5
U	2	4
V	8	2
W	8	2
X	16	1.5
Y	8	2
Z	20	1.5

Pelistä löytyy myös ostettavia tehosteita, jotka vaikuttavat pelin kulkuun. Tehosteisiin vaadittavia kolikoita voi ansaita pelaamalla tai ostamalla rahalla.



KUVIO 1. WordCraftin pelitila

2 ALUSTARIIPPUMATON KEHITYS

2.1 Määritelmä

Alustariippumattomalla kehityksellä tarkoitetaan ohjelmistoa, joka toimii usealla eri laitteistoalustalla tai käyttöjärjestelmällä mahdollisimman samalla tavalla. Tämä tarkoittaa, että ohjelmiston kehityksessä ei käytetä laitteistoalustaan tai käyttöjärjestelmään sidoksissa olevia ominaisuuksia tai ohjelmointikieltä. Tärkeimpinä alustariippumattomuuden piirteitä ovat mm. lukujen esitysmuoto, grafiikkakirjastot, useimmat järjestelmäkutsut sekä säikeiden ja prosessien toteutus. Alustariippumattomuudella ei kuitenkaan tarkoiteta, että ohjelmisto toimisi kaikilla mahdollisilla alustoilla ja käyttöjärjestelmillä. Useasti ohjelmat kehitetäänkin vain muutamille, suosituimmille alustoille ja käyttöjärjestelmille. (Alustariippumattomuus 2012.)

Alustariippumattomuus on yleistynyt vuosien saatossa. Ennen ohjelmat olivat oletustusti käyttöjärjestelmäriippuvaisia sekä yleensä sidottuja tietokonearkkitehtuuriin, jopa vain tiettyyn tietokonemalliin. Merkittävimpänä tekijänä alustariippumattomuuden nopeaan yleistymiseen pidetään WWW:n nopeaa laajenemista ja laajaa saatavuutta. (Mt.) Kuviossa 2 on esimerkki alustariippuvaisesta ohjelmakoodista ja kuviossa 3 alustariippumattomasta ohjelmakoodista.

```
# sidottu alustaan, koska hakemistoa /proc/sys/loadavg ei löydy koskaan Windows- tai Mac-järjestelmistä
def get_system_load()
    info_path = '/proc/sys/loadavg'
    load_avg0 = get_current_loads(info_path)
    now = time.time()
    sys.sleep(1)
    load_avg1 = get_current_loads(info_path)
    elapsed = time.time()-now
    return load_avg1-load_avg0/elapsed
```

KUVIO 2. Pseudokoodiesimerkki alustariippuvaisesta ohjelmakoodista. (Alustariippumattomuus 2012)

```
# laitteistoalustariippumaton, käyttää kielen omaa funktiota, joka on
# toteutettu erikseen eri käyttöjärjestelmille
def get_system_load()
    load_avg0 = system.loadavg()
    now = time.time()
    sys.sleep(1)
    load_avg1 = system.loadavg()
    elapsed = time.time()-now
    return load_avg1-load_avg0/elapsed
```

KUVIO 3. Pseudokoodiesimerkki alustariippumattomasta ohjelmakoodista. (Alustariippumattomuus 2012)

2.2 Hyödyt ja haasteet

Alustariippumattoman kehityksen hyötyjä on monia. Yleisesti tärkeimpänä hyötynä pidetään tehokkuutta. Kun ohjelmakoodi on kirjoitettu, se toimii kaikilla kohdealustoilla, eikä jokaiselle alustalle tarvitse kirjoittaa omaa ohjelmakoodia. (Why cross platform 2013.)

Toinen tärkeä hyöty on oman kehityksessä käytetyn ohjelmointikielen osaaminen. Mikäli kaikille alustoille kirjoittaisi omat ohjelmat ilman yhteistä, alustariippumatonta kieltä ja tekniikoita, tulisi ohjelmoijan opetella mm. Androidille Java, iOS:lle Objective-C ja niin edelleen. (Mt.)

Kolmantena hyötynä on kehittämisaika ja sen ylläpidettävyys. Koska alustariippumattomalle ohjelmalle yksi ohjelmakoodi yhdellä kielellä on riittävä, lyhenevät ohjelmakoodin kirjoittamiseen tarvittava aika ja ylläpidon helppous ja selkeys huomattavasti. (Mt.)

Alustariippumattomassa kehityksessä on myös haasteita. Alkuun pääseminen on hidas prosessi ja vaatii paljon työtä, jotta sama ohjelmakoodi saadaan edes kääntymään halutuille alustoille. Se vaatii myös paljon perehtymistä ja tietynlaisen ajattelutyylin opettelua, jotta oppii tietämään ja ymmärtämään, mitkä toiminnot ovat alustariippuvaisia ja mitkä alustariippumattomia. Edellä mainitun hitaan alun ja vaadittavan perehtymismäärän vuoksi monet kehittäjät suosivat kolmannen

osapuolen ohjelmistokehityspaketteja (third party SDK) alustariippumattomassa kehityksessä. (Mt.)

Vaikka ohjelmistokehityspaketit helpottavatkin alkuun pääsemisessä, tuovat ne mukanaan myös omat haasteensa. Koska ohjelmistokehityspaketti tulee kolmannelta osapuolelta, se tekee kehittäjästä riippuvaisen tästä osapuolesta. Myös joitain haluttuja toimintoja ja ominaisuuksia voi puuttua ohjelmistokehityspaketista, ja uusien ominaisuuksien toteutuksessa ja ohjelmistokehityspaketin päivityksessä voi kestää kauan aikaa. (Mt.)

2.3 Käytetyt alustat ja niiden eroavaisuudet

Tuetut pelialustat ja niissä käytetyt ohjelmointikielet:

- Apple iOS – C++
- Android – C++
- Samsung Bada – C++
- Symbian – C++
- BlackBerry OS 10 – C++
- Meego – C++
- Maemo – C++
- Windows Phone 7 – C#
- BlackBerry Java, OS 7.x - Java
- Nokia S40 - Java
- Web / Facebook - Java
- Windows Desktop – C++
- Linux Desktop – C++
- Apple OSX-Desktop – C++

- Windows 8 (kehityksessä)

Kaikkiin alustoihin, pois lukien Windows Phone 7, voidaan kehittää pelejä myös Lua-ohjelmointikieltä käyttäen. (Supported Platforms 2012.)

Eroavaisuuksia alustoista löytyy mm. käytetyistä ohjelmointikielistä, suorittimien arkkitehtuurien käyttäytymisessä (mm. lukujen pyöristäminen), näyttöjen kuvasuhteissa ja koossa, tekstuureille varatussa muistissa, käyttöjärjestelmärajoitteissa ja monissa muissa asioissa. (Challenges of Cross Platform Game Development – part 1-3 2009.)

3 MENETELMÄT JA TYÖKALUT

3.1 Ohjelmointikieli

Toteutuksen ohjelmointikielenä käytettiin C++:aa.

C++ on pitkälle vuosien varrella kehittynyt olio-ohjelmointia tukeva ohjelmointikieli. Se kääntyy useimmille olemassa oleville mobiilialustoille mm. vakiokirjaston siirrettävyyden ansiosta ja täten sopii erinomaisesti alustariippumattomaan kehitykseen. C++ on periaatteellisesti parempaa C:tä ja tukee olio-ohjelmoinnin lisäksi tiedon abstrahointia sekä geneeristä ohjelmointia. Ohjelmointikieli on suunniteltu mahdollisimman yleiskäyttöiseksi ja täten ylimääräiset suoritusenaikaset koodit, mm. roskienkeräys, on jätetty pois (Stroustrup 2000, 7-8 ja 21–22).

3.2 Työkalut

Työkaluina ohjelmoinnin toteutuksessa toimivat Visual Studio 2010, ActivePerl ja DieselMediaPacker.

Visual Studio on Microsoftin kehittämä kehitystyökalu, joka tukee useita eri ohjelmointikieliä. Tuettuihin ohjelmointikieliin kuuluvat mm Visual Basic, C++, C# ja J#.

Visual Studiolla voidaan tehdä monenlaisia sovelluksia, kuten mm. web-, mobiili- ja Windows-sovelluksia. Myös graafisten käyttöliittymien luominen onnistuu kattavien työkalujen avulla. Visual Studio 2010 versio julkaistiin 12.4.2010, ja siinä on monia uudistuksia edeltäjäänsä Visual Studio 2008. (Visual Studio n.d.)

ActivePerl suorittaa Perl-ohjelmointikielen skriptoja. Perl-skriptoja käytetään Star Arcadella mm. grafiikkatiedostojen skaalaamiseen ja pelissä käytettyjen resurssitiedostojen pakkaamiseen DieselMediaPackerin avulla.

DieselMediaPacker on Star Arcaden kehittämä ja käyttämä ohjelma pelissä käytettyjen tiedostojen pakkaamiseen. Pakattuihin tiedostoihin kuuluvat mm. kuvat, teemat, sanakirja, xml-tiedostot, fontit ja niin edelleen. DieselMediaPacker pakkaa tiedostot dmp-päätteiseksi ja samalla suojaa ulkopuolisten asiattoman pääsyn tiedostoihin.

3.3 OpenGL ja OpenGL ES

OpenGL muodostuu sanoista Open Graphics Library ja se on laitteistoriippumaton ohjelmointirajapinta. OpenGL:llä toteutetaan graafisia toimintoja noin 250:llä eri funktiolla, ja se on suosittu erityisesti peliohjelmointialalla. Sitä voidaan käyttää kaksi- ja kolmiulotteisten kappaleiden piirtämiseen. Pahimpana kilpailijana OpenGL:lle on Microsoftin Direct3D. (OpenGL Overview n.d; OpenGL 2012.)

OpenGL itsessään on vain pelkkä grafiikkakirjasto. Muut asiat kuten käyttäjän syötteet, ikkunoiden luominen ja median hallitseminen tehdään yleensä lisärajapinnoilla kuten esim. SDL (Simple Direct media Layer), GLUT (Graphics Library Utility Toolkit) ja GLFW (GL Frame Work) (ks. kuvio 4). (OpenGL Overview n.d; OpenGL 2012.)

```

#include <GL/gl.h>
#include <GL/glut.h>
#include <GL/glu.h>

void display() { /* empty function    required as of glut 3.0 */ }

int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowSize(800,600);
    glutCreateWindow("Hello World");
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

KUVIO 4. Ikkunan alustaminen GLUT rajapintaa käyttäen (OpenGL Programming/Basics/Structure 2012)

OpenGL:n kehitystyö tapahtuu Khronos Groupin alaisuudessa. Se päivittyy aktiivisesti, ja siihen lisätään uusia ominaisuuksia ja funktioita päivitettyjen versioiden myötä. Tuorein versio, 4.3, julkaistiin 6.8.2012. (OpenGL 2012.)

OpenGL ES on sulautettuja järjestelmiä varten tehty toteutus OpenGL:stä. Sitä käytetään mm. mobiililaitteissa ja pelikonsoleissa, ja se toteuttaa suurimman osan OpenGL:n toiminnollisuudesta. Erojakin myös löytyy mm. syntaksista ja piirtotekniikoista. OpenGL ES:ää tukevat mm. Android, Apple, ARM, Qualcomm, Marvell, Intel, NVIDIA, NOKIA OYJ ja monet muut. (OpenGL ES - The Standard for Embedded Accelerated 3D Graphics n.d.)

4 TOTEUTUS JA TYÖPROSESSI

4.1 Aikataulu

Aikataulun seurantaan käytettiin BaseCamp -projektinhallintaohjelmaa. BaseCamp-ohjelmaan kirjattiin työpäivän päätteeksi työtunnit ja tehtävät. WordCraftin toteutus alkoi 27.4.2012 ja jäi tauolle 10.8.2012 sisältäen 412.5 tuntia eli 55 työpäivää.

Toteutusaikaan kuuluu itse pelin toteuttamisen lisäksi pieni määrä pelitestausta ja pelin tasapainoin hiomista, lisäksi jotkut ominaisuudet tehtiin uudelleen kehityksen edetessä.

Suurimmat yksittäiset ajankäytöt olivat kirjaimille tarkoitettu sanapalkki ja sen toiminnollisuus, sanakirja ja sen hakualgoritmit ja toiminnollisuus, pisteiden lasku, pisteruutu ja pelin päättymislogiikka, tekoälyn tekeminen sekä pelin käytettävyyden ja pelattavuuden hiominen ja testaaminen.

4.2 StarArcadeClient

StarArcadeClient on StarArcaden peleissä käytetty sosiaalinen moninpelikerros. Se sisältää käyttöliittymän ja toiminnollisuuden peleissä esiintyvään tähtivalikkoon, kirjautumisikkunaan sekä pelaajien väliseen keskusteluun.

Kirjautumisikkuna pyytää käyttäjää joko rekisteröitymään StarArcaden pelitilin, joka käy kaikissa Star Arcaden peleissä, tai kirjautumaan sisään vieraana. Mikäli käyttäjä päättää rekisteröityä, hän voi tehdä sen antamalla haluamansa käyttäjätunnuksen ja salasanan sekä sähköpostiosoitteen varmistusta varten. Myös Facebook-tunnuksilla kirjautuminen ja rekisteröityminen ovat mahdollisia.

Tähtivalikko sisältää neljä kohtaa: minä, muut pelaajat, esineet ja yleiset asetukset. ”Minä”-valikosta löytyy tietoja pelaajan käyttäjätunnuksesta, pelaajakuvasta, pelatuista peleistä, voittoprosentista, pisteistä sekä virtuaalisen valuutan määrästä. Valikosta voi myös ostaa lisää valuuttaa tai kutsua kavereita rekisteröitymään ja pelaamaan vaikkapa Facebookin kautta. ”Muut pelaajat”-valikosta löytyvät linjoilla olevat pelaajat sekä kaverilistat. Valikosta voit myös haastaa toisia pelaajia pelaamaan tai lähettää kavereille lahjaksi esineitä. ”Esineet”-valikossa ovat virtuaalisella valuutalla myytävät erikoisesineet. Erikoisesineitä ovat mm. uudet aseet, pistetehosteet ja teemat. Myös jo ostetut esineet löytyvät valikon alta. ”Yleiset asetukset” kattavat valitun kielen, peliäänet ja -musiikin, tärinän sekä ilmoitusten määrän.

Toteutettaessa StarArcadeClientin vaatimat rajapinnat saadaan em. ominaisuudet kehitettävän pelin käyttöön ja pelinkehittäjä voi keskittyä itse pelin kehittämiseen. Tämän vuoksi pelin toteutus yleensä alkaakin pelitilalla, joka toteuttaa vaaditut sekä kehittäjän haluamat rajapinnat StarArcadeClientista. Toteutettuaan vaadittavat asiat voi pelitilasta yksinkertaisilla rajapintakutsuilla esimerkiksi laittaa käyttäjän pelijonoon etsimään vastustajaa muista samalla pelitunnuksella olevista pelaajista.

4.3 DieselEngine

DieselEngine on StarArcadella käytetty pelimoottori. Pelimoottorilla yleisesti tarkoitetaan pelien ohjelmarunkoa, joka kattaa pelimekaniikat mm. pelin grafiikan piirtämisen näytölle sekä fysiikan mallinnuksen. DieselEngine toteuttaa tiedostojen ja graafisten tekstuurien lataamisen lisäksi useita OpenGL, OpenGL ES sekä DirectX-funktioita ja toiminnollisuutta. Se toteuttaa myös Box2D-fysiikan mallinnuksen. DieselEngine vastaa myös laitteistolta tulevista syötteistä mm. sormen painalluksesta ja pyyhkäisystä ja toimittaa ne edelleen kehittäjän pelitilalle.

DieselEnginen piirtofunktioita pääsee kutsumaan esimerkiksi pyytämällä rajapinnasta valmiiksi alustetun grafiikkaosan, joka toteuttaa DieselEnginen piirtofunktioiden rajapinnat.

DieselEnginen monipuolisen toteutuksen ansiosta on mahdollista saada pelimoottori kääntymään usealla eri alustalle peleineen samalla ohjelmakoodilla. Sen rajapinnat ovat selkeästi toteutettu, eikä alustariippumattomuus sekä useat eri toteutetut piirtorajapinnat tule pelin kehittäjälle esille tai rajoitteiksi.

4.4 WordCraft

WordCraft toteuttaa StarArcadeClientin vaatimat rajapinnat pelitilassaan. Se käyttää myös DieselEnginen piirtofunktioita grafiikan piirroksessa. WordCraftin pelitilan koodirakenne menee kuvion 5 mukaisesti.

```

/**
 * Startup
 * Kutsutaan, kun käyttäjä on kirjautunut sisään.
 * Alustetaan muuttujat ja ladataan resurssit.
 */
void Startup();

/**
 * Shutdown
 * Kutsutaan, kun ohjelma suljetaan.
 * Tuhotaan muuttujat ja puretaan ladatut resurssit.
 */
void Shutdown();

/**
 * Draw
 * Kutsutaan automaattisesti jokaisella piirtokierroksella.
 * Funktiossa piirretään kaikki piirrettävät asiat.
 */
void Draw();

/**
 * Kutsutaan automaattisesti jokaisella päivityskierroksella.
 * Päivittää kaikki olemassa olevat objektit ja ajastimet.
 */
void Update();

/*
 * RecvNetworkMessages
 * Kutsutaan automaattisesti, kun verkkoviesti saapuu.
 * Käsitellään verkkoviestin mukana tullu tieto.
 */
void RecvNetworkMessages();

/*
 * OnMouseButton
 * Kutsutaan, kun hiiren tai sormen painallus tapahtuu.
 * Käsitellään painalluksesta tapahtuvat asiat.
 */
void OnMouseButton();

/*
 * OnMouseDown
 * Kutsutaan, kun hiiren tai sormen pyyhkäisy tapahtuu
 * Käsitellään pyyhkäisystä tapahtuvat asiat.
 */
void OnMouseDown();

/*
 * OnMouseButtonUp
 * Kutsutaan, kun hiiri tai sormi nostetaan pois näytöltä
 * Käsitellään sormen nostosta tapahtuvat asiat
 */
void OnMouseButtonUp();

/**
 * OnKeyDown
 * Kutsutaan, kun näppäimen painallus tapahtuu
 * Käsitellään kyseisen näppäimen painalluksesta tapahtuvat asiat
 */
void OnKeyDown();

/**
 * StartGame
 * Kutsutaan, kun halutaan, että peli alkaa
 * Hoitaa pelin alkamiseen vaaditut alustamiset ja pelitilan asettamisen.
 */
void StartGame();

/**
 * OnGameEnded
 * Kutsutaan, kun peli loppuu
 * Hoitaa pelin aikana käytettyjen tietojen ja resurssien tuhoamisen
 */
void OnGameEnded();

/**
 * GameDisconnect
 * Kutsutaan, jos peliyhteys katkeaa odottamattomasti
 * Käsitellään yhteyden katkeamisesta johtuvat tapahtumat
 */
void GameDisconnect();

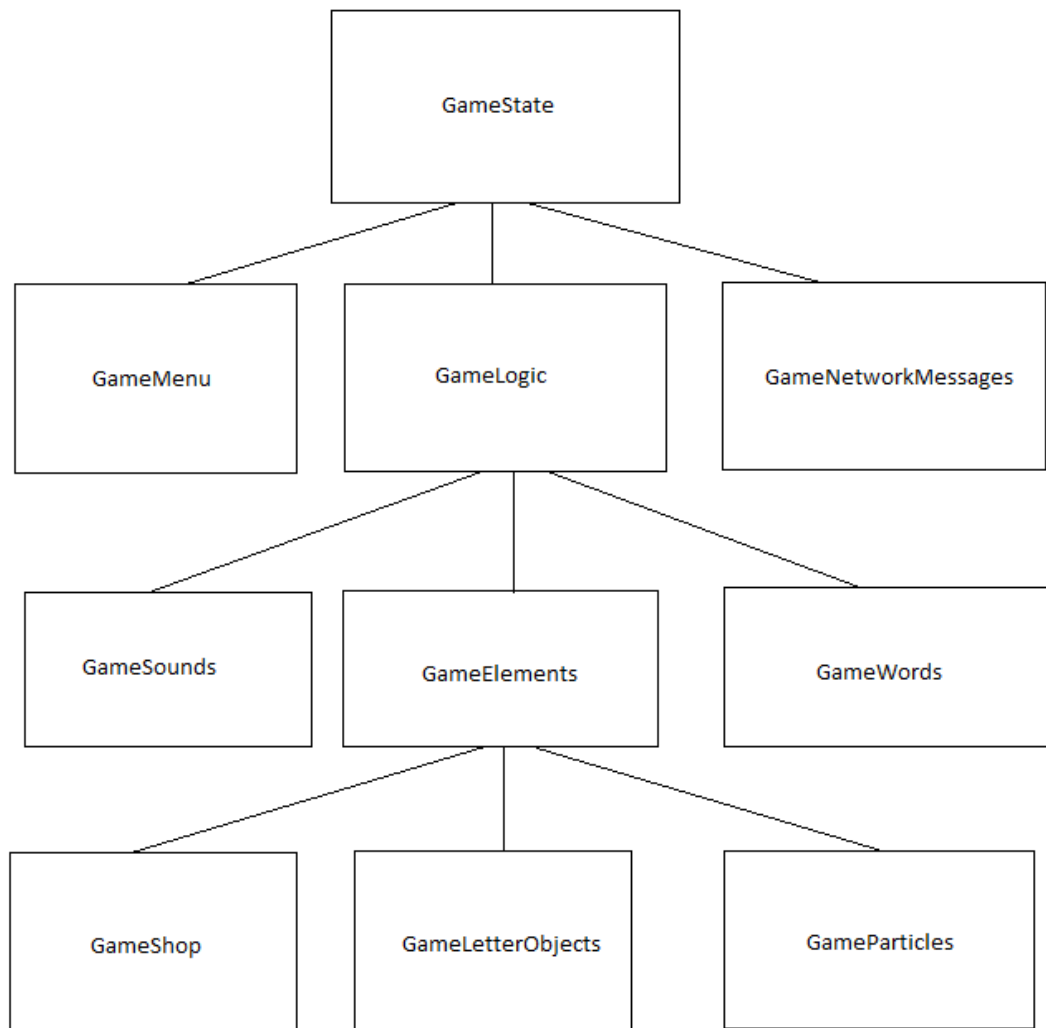
```

KUVIO 5. Pelitilan rakenteen koodiesimerkki

Pelitila omistaa myös useita luokkia. Luokkiin kuuluvat GameElements, GameLogic, GameSounds, GameNetworkMessages, GameLetterObjects, GameWords, GameMenu, GameShop ja GameParticles.

- GameElements-luokka sisältää kaikki pelissä olevat oliot sekä hoitaa olioiden piirtämisen DieselEnginen piirtofunktioilla.
- GameLogic-luokka sisältää pelissä käytetyn logiikan sekä huolehtii olioiden päivittämisestä.
- GameSounds-luokka sisältää pelissä käytetyt äänet ja huolehtii niiden lataamisesta sekä äänien soittamisesta.
- GameNetworkMessages-luokka sisältää tietorakenteet lähetettävälle verkkoviesteille.
- GameLetterObjects-luokka on kerroin- ja kirjainolioiden muodostamiseen tarkoitettu. Sisältää myös olioiden toiminnollisuuden.
- GameWords-luokka sisältää sanakirjan ja sanalistat. Huolehtii myös kirjainten löytämisestä ja sanakirjan lataamisesta.
- GameMenu-luokka pitää yllä pelin valikoiden ja nappien tilan ja huolehtii pelin valikoiden syötteistä.
- GameShop-luokka huolehtii esineiden ostosta pelin aikana. Se pitää yllä myös esineiden tiedot, lukumäärät, tyypit ja hinnat.
- GameParticles-luokka on partikkeliolioiden muodostamista varten. Sisältää olioilta vaaditut ominaisuudet ja toiminnollisuudet.

Luokat noudattavat luokkahierarkiaa, jossa pelitilaluokka GameState on ylimpänä (ks. kuvio 6). GameState käyttää luokkia GameMenu, GameNetworkMessages ja GameLogic. GameLogic käyttää luokkia GameSounds, GameWords ja GameElements. GameElements taas käyttää luokkia GameShop, GameLetterObjects ja GameParticles.



KUVIO 6. WordCraftin luokkahierarkia

WordCraftin keskeisenä osana toteutusta on sanakirja ja kirjainten löytäminen. Peli ottaa käynnistyessä vastaan estettyjen ja lisättyjen sanojen listat palvelimelta ja lataa sanakirjan sekä listat muistiinsa. Tämä mahdollistaa helpon ja nopean tavan hallita pelin hyväksymiä sanoja.

Kirjainten löytäminen sanankirjasta on toteutettu karsimalla kaikki kolmella ensimmäisellä kirjaimella alkavat sanat omaan taulukkoonsa. Kirjaimia poistettaessa tai lisättäessä peli käy taulukon sanat läpi ja tarkastaa löytyykö vastaavaa sanaa taulukosta.

4.5 Tekoäly

WordCraftissa voi pelata verkkopelaamisen lisäksi tekoälyä vastaan. Tekoälyn suorittaminen tapahtuu StarArcaden palvelimilla.

Tekoälyllä on sama englanninkielinen sanakirja ja identtinen kirjainkasa kuin pelaajalla. Sanakirjasta on poistettu kiro sanat hämmennyksen ja solvausten poistamiseksi.

Tekoäly ratkaisee pelin alkaessa annetusta kirjankasasta asetetun määrän sanoja valmiiksi annetun sanan pituuden rajojen mukaan. Sanojen lukumäärää ja sanan pituuden rajaa voidaan muuttaa tekoälyn vaikeusasteen säätämiseksi.

Sanan ratkaiseminen on toteutettu samalla algoritmilla kuin itse pelin toteutuksessa. Tekoäly etsii annetusta kirjainkasasta kolme satunnaista kirjainta ja tarkastaa, kuinka monta kirjainyhdistelmällä alkavaa sanaa sanakirjasta löytyy. Mikäli sanoja on tarpeeksi, ottaa tekoäly satunnaisen sanan kirjainyhdistelmällä alkavista sanoista ja alkaa etsimään puuttuvia kirjaimia kirjainkasasta. Sama prosessi toistetaan, kunnes tekoälyllä on määritetty määrä sanoja valmiina.

Tekoäly jakaa pelin ajan valmiiksi löydettyjen sanojen määrällä, jotta se osaa merkata sanat ”löydetyiksi” sopivan ajan välein pelin kulun aikana. Mikäli löydetty sana on tarpeeksi pitkä, lähettää tekoäly vastustajalle kuuluvat lumihiutaleet. Lumihiutaleiden määrää voidaan tarvittaessa vähentää käytettävyyden ja pelikokemuksen parantamiseksi.

Kertoimia tekoäly käyttää aina kun mahdollista, eli kun edellinen sana on ollut tarpeeksi arvokas kertoimen saamiseksi.

5 POHDINTA

5.1 Johtopäätökset

WordCraft on ollut työnä mielenkiintoinen ja haastava. Peliä kehittäessä, suunnitella ja testatessa on oppinut paljon uutta pelinkehityksestä alana. Pelin kehittäminen on kovaa ja haastavaa työtä. Ala on alati muuttuva ja pelaajien tarpeet vaativia. Joskus itsestään selviltä tuntuvat asiat aiheuttavat eniten miettimistä tai yksinkertaiselta vaikuttavat vaatimukset ovatkin asioita mihin uppoaa eniten aikaa ja vaivaa.

Myös tekninen osaaminen on karttunut pelin kehityksen aikana. Peli on tuonut uutta näkökulmaa ja ajattelumallia pelin ohjelmoinnin toteutukseen. Jos peli pitäisi tehdä jälkikäteen uudelleen, tulisi ohjelmistokoodista rakenteeltaan ja luettavuudeltaan selkeämpää ja parempaa.

Vaikka peleille ei varsinaista onnistumiskaavaa menestykseen ole, ja joskus hyvätkin pelit eivät saa ansaitsemaansa suosiota ja huomiota, WordCraftin uskotaan onnistuvan julkaisussaan. Sen takana on osaava ryhmä ollut ideoimassa, testaamassa, hio-
massa ja kehittämässä peliä, eikä kehityksen tai julkaisun kanssa ole liikaa kiirehditty.

Kaikki ei kuitenkaan ole mennyt täydellisesti kehityksen aikana. Projektin aikana monesti asioita muutettiin ja tehtiin uudelleen, mikä toi pelille ylimääräistä kehittämis-aikaa. Myös ohjelmakoodin suunnitellu olisi voinut olla tarkempaa muutamien ominaisuuksien kannalta, jotta vaatimusten muuttuessa olisi ominaisuuksiakin ollut helppompaa muuttaa, poistaa ja lisätä.

5.2 Tulevaisuus

WordCraftin nimi muuttui dokumentoinnin aikana ja se tullaan julkaisemaan todennäköisesti vuoden 2013 ensimmäisellä vuosineljänneksellä. Peli kuitenkin päivitetään Star Arcaden kolmannen sukupolven sosiaalisen kehikseen ennen pelin julkaisua.

Myös markkinoinnille ja pelin esittelylle on varattu oma aikansa ennen julkaisun tapahtumista.

Pelistä muutetaan myös monta muuta asiaa. Esimerkiksi pelin sisäinen ostovalikko, alkuvalikot ja sivuvalikko tehdään kokonaan uudestaan.

Myös pelin käyttöliittymä, tasapaino ja kirjainten pisteytykset muokataan, ja pelille luodaan uuden nimen mukainen teema.

Peliin toteutetaan myös ohjattu pelitila, jossa käyttäjää opastetaan, kuinka peliä pelataan.

Julkaisun jälkeen peliä ylläpidetään ja päivittämään tarpeen mukaan. Myös mahdolliset löydetyt ohjelmointivirheet korjataan.

LÄHTEET

Alustariippumattomuus. 2012. Wikipedia – vapaa tietosanakirja. Viitattu 20.1.2013.
<http://fi.wikipedia.org/wiki/Alustariippumattomuus>

Challenges of Cross Platform Game Development – Part 1-3. 2009. YouTube video-palvelussa. Viitattu 14.12.2012. <http://www.youtube.com/watch?v=9q94YIFCVBU>

Company presentation. 2012. Esite. Star Arcaden markkinointimateriaali.

OpenGL ES - The Standard for Embedded Accelerated 3D Graphics. n.d. Viitattu 6.12.2012. <http://www.khronos.org/opengles/>

OpenGL Overview. n.d. Viitattu 6.12.2012. <http://www.opengl.org/about/>

OpenGL Programming/Basics/Structure. 2012. Wikikirjasto – vapaita kirjoja avoimelle maailmalle. Viitattu 5.12.2012.
http://en.wikibooks.org/wiki/OpenGL_Programming/Basics/Structure

OpenGL. 2012. Wikipedia – vapaa tietosanakirja. Viitattu 5.12.2012.
<http://fi.wikipedia.org/wiki/OpenGL>

Stroustrup, B. 2000. C++-ohjelmointi. Vantaa: Tummavuoren Kirjapaino Oy.

Supported Platforms. 2012. Esite. Star Arcaden markkinointimateriaali.

Visual Studio. n.d. Viitattu 6.12.2012. [http://msdn.microsoft.com/en-us/library/52f3sw5c\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/52f3sw5c(v=vs.100).aspx)

Why cross platform. 2013. Julkaisu Cross Platform Developmentin sivustolla. Viitattu 20.1.2013. <http://cross-platformdevelopment.com/why-cross-platform/>